

10

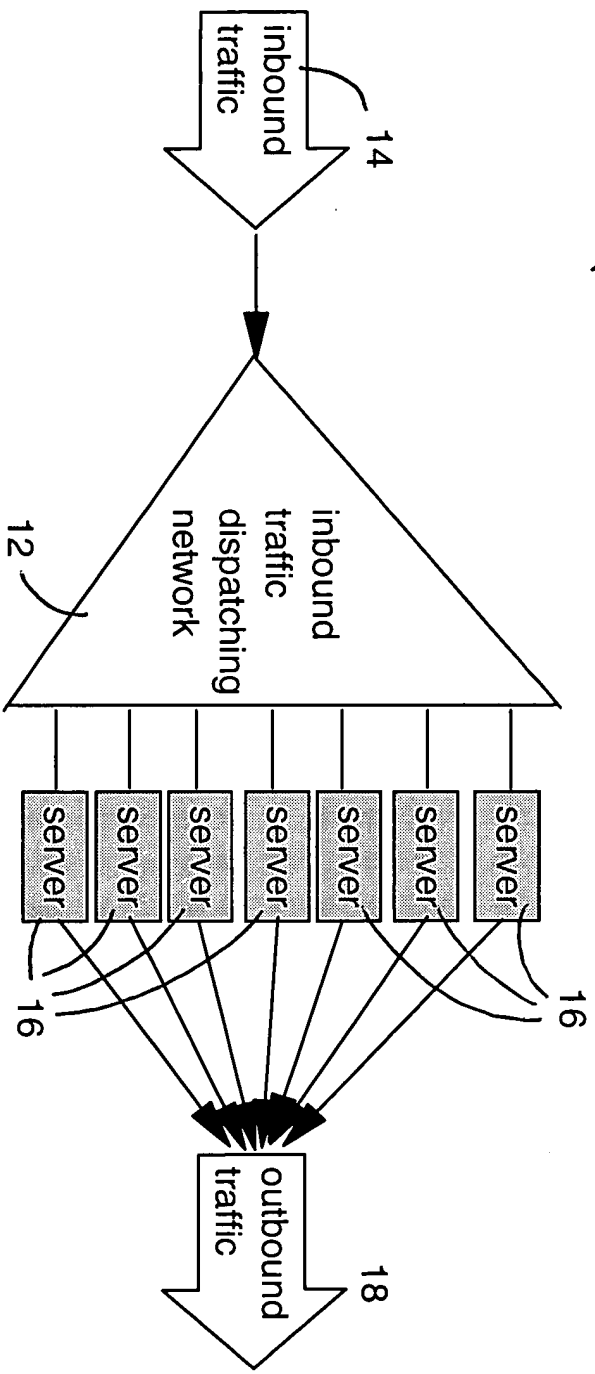
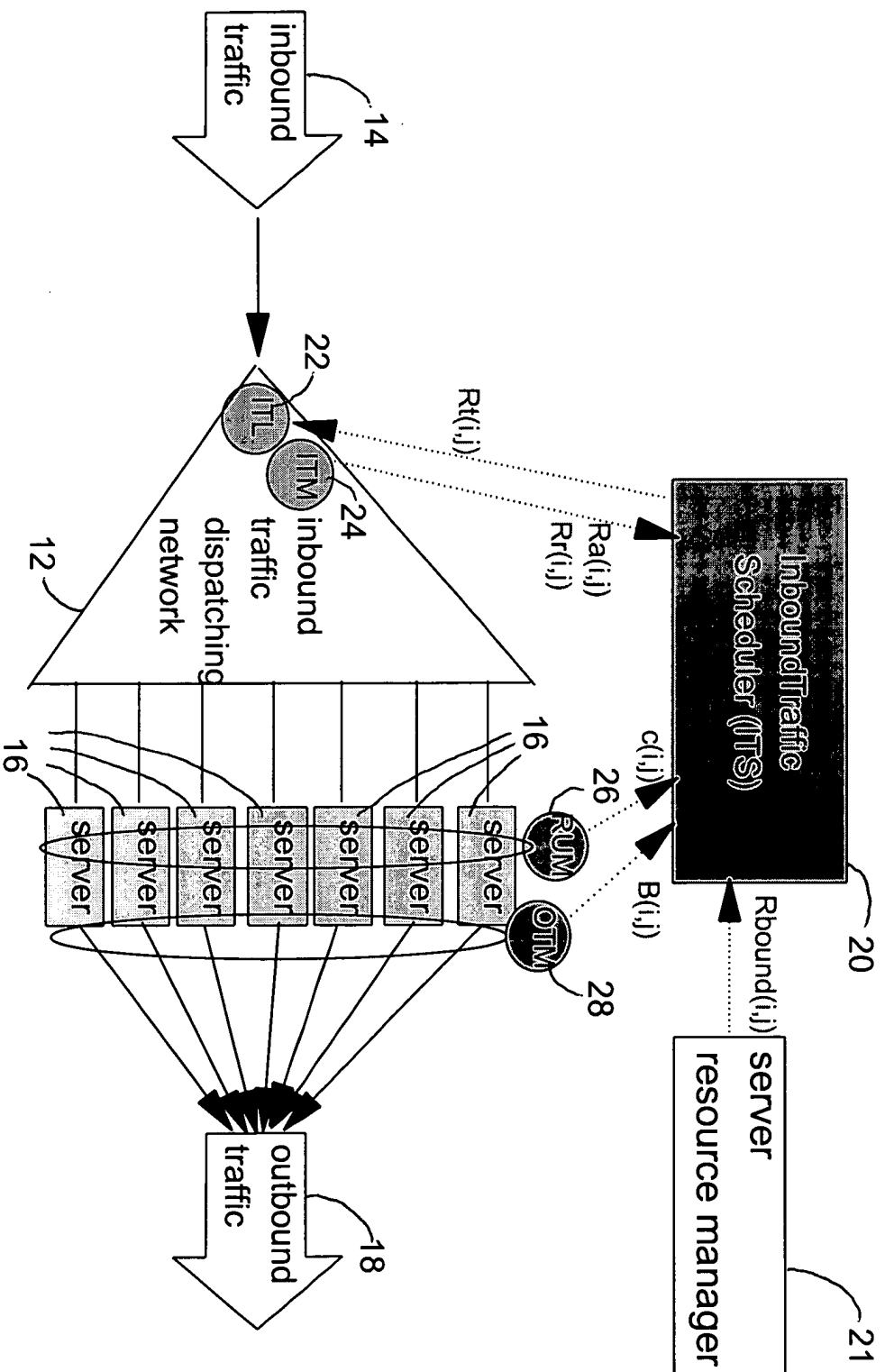


Fig. 1.

Fig. 2.



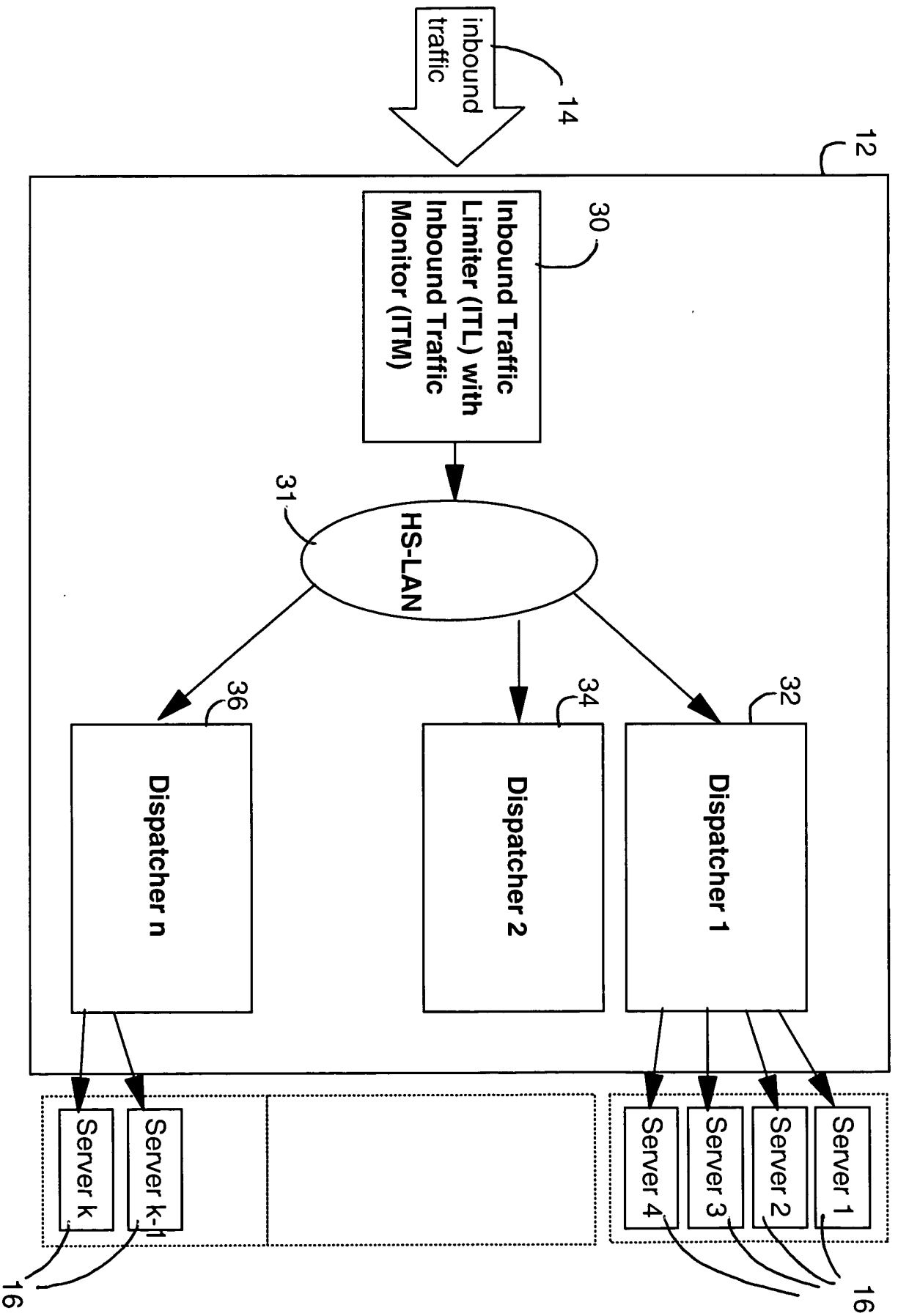


Fig. 3

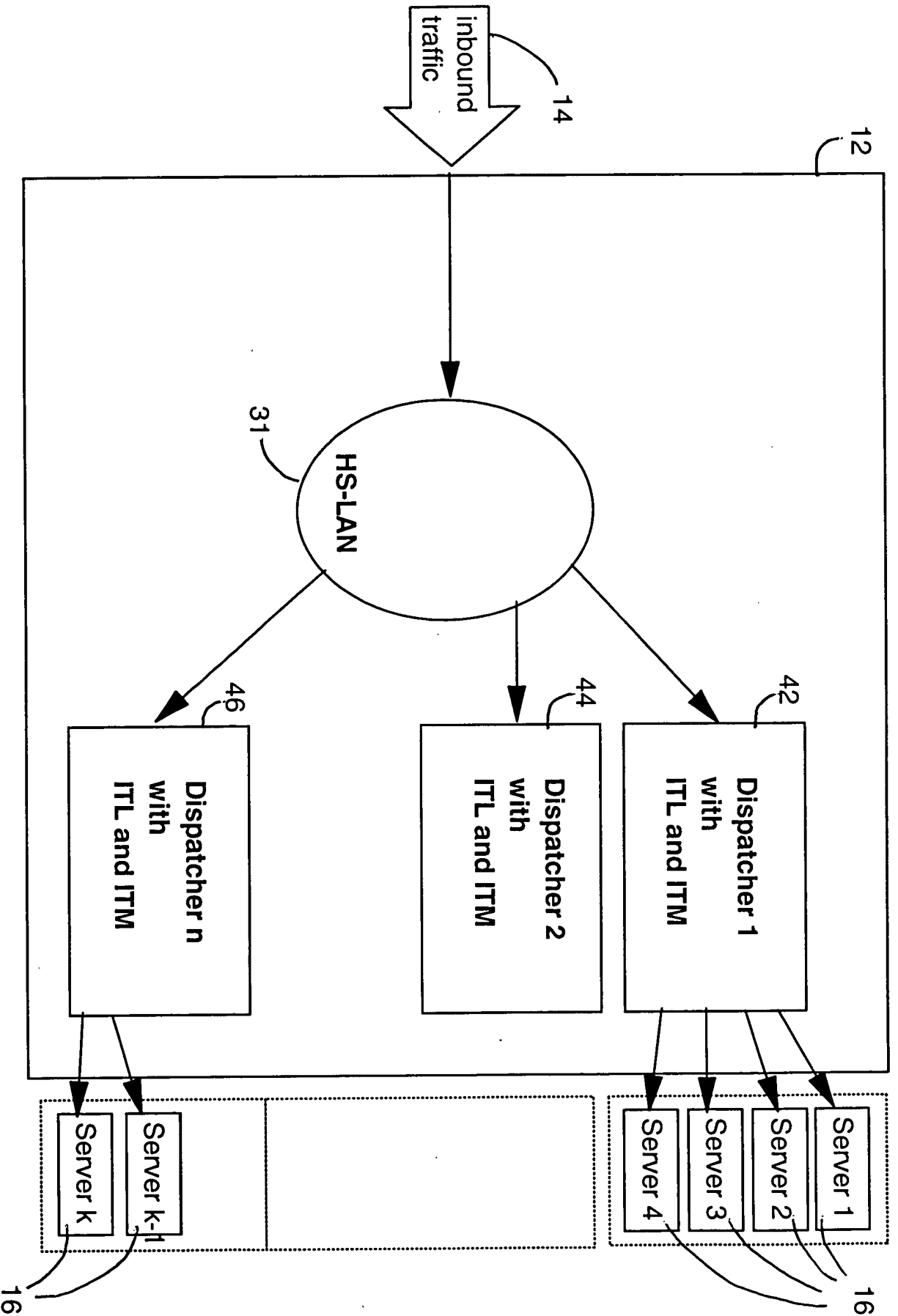
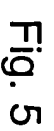


Fig. 4



step 1

```

START:
For all (i,j), set Bmax(i,j) = min (Bmax(i,j),Rbound(i,j)*b(i,j)); /* reflecting "external constraint" Bbound(i,j) */
For all (i,j), set Bt(i,j) = b(i,j)*Ra(i,j); /* estimating the current outbound traffic */
Let Bt be the sum of Bt(i,j) over all (i,j); /* estimating the current total outbound traffic */
if Bt>Btotal then go to step 2; /* link congestion detected */
if ((Rr(i,j)=0) /* no request rejection */
&(Bt(i,j)<=Bmax(i,j))) over all (i,j) /* SLA is not violated. "<=" means less-than-or-equal-to */
then go to step 5; else go to step2;

```

61

step 2

```

COMPUTE_BANDWIDTH_TARGETS:
For all (i,j), set Bt(i,j) = b(i,j)*(Ra(i,j)+Rr(i,j)); /* computing new targets for bandwidth usage */
Let Bt be the sum of Bt(i,j) for all (i,j); /* estimating outbound traffic when all requests are admitted */
For every (i,j) such that Bt(i,j)>Bmax(i,j) /* this step is needed since Bt(i,j) were just re-computed */
first set Bt = Bt-(Bt(i,j)-Bmax(i,j)) /* wants to generate more than the maximum SLA */
and then set Bt(i,j) = Bmax(i,j); /* adjusting expected total outbound traffic */
if Bt<=Btotal /* bounding traffic by maximum SLA */
then go to step t; else go to step 3; /* no link congestion will be anticipated */

```

62

step 3

```

Let Bexcess be the sum of (Bt(i,j)-Bmin(i,j)) over those Bt(i,j)>Bmin(i,j); /* computing "excess" bandwidth */
/* perform either Case 1 or Case 2 */
/* Case 1: compute "sharable" bandwidth when bandwidth borrowing is permitted */
Let Bsharable be Btotal minus the sum of smaller of (Bt(i,j) and Bmin(i,j)) over all (i,j);
/* Case 2: compute "sharable bandwidth when bandwidth borrowing is not permitted */
Let Bsharable be Btotal minus the sum of Bmin(i,j) over all (i,j);
For every (i,j) such that Bt(i,j)>Bmin(i,j) /* perform fair proration */
set Bt(i,j) = Bmin(i,j) + (Bt(i,j)-Bmin(i,j))*(Bsharable / Bexcess);

```

63

step 4

```

COMPUTE_NEW_RATES:
For every (i,j) such that Bt(i,j)<=Bmin(i,j) set Bt(i,j) = Bmax(i,j); /* this is equivalent to "no throttling" */
For every (i,j) set Rt(i,j) = Bt(i,j) / b(i,j); /* computing target rates */
Optionally compute Rt(i,j,k) from Rt(i,j) for all k; /* optional computation, rate for each server */

```

64

step 5

STOP:

65



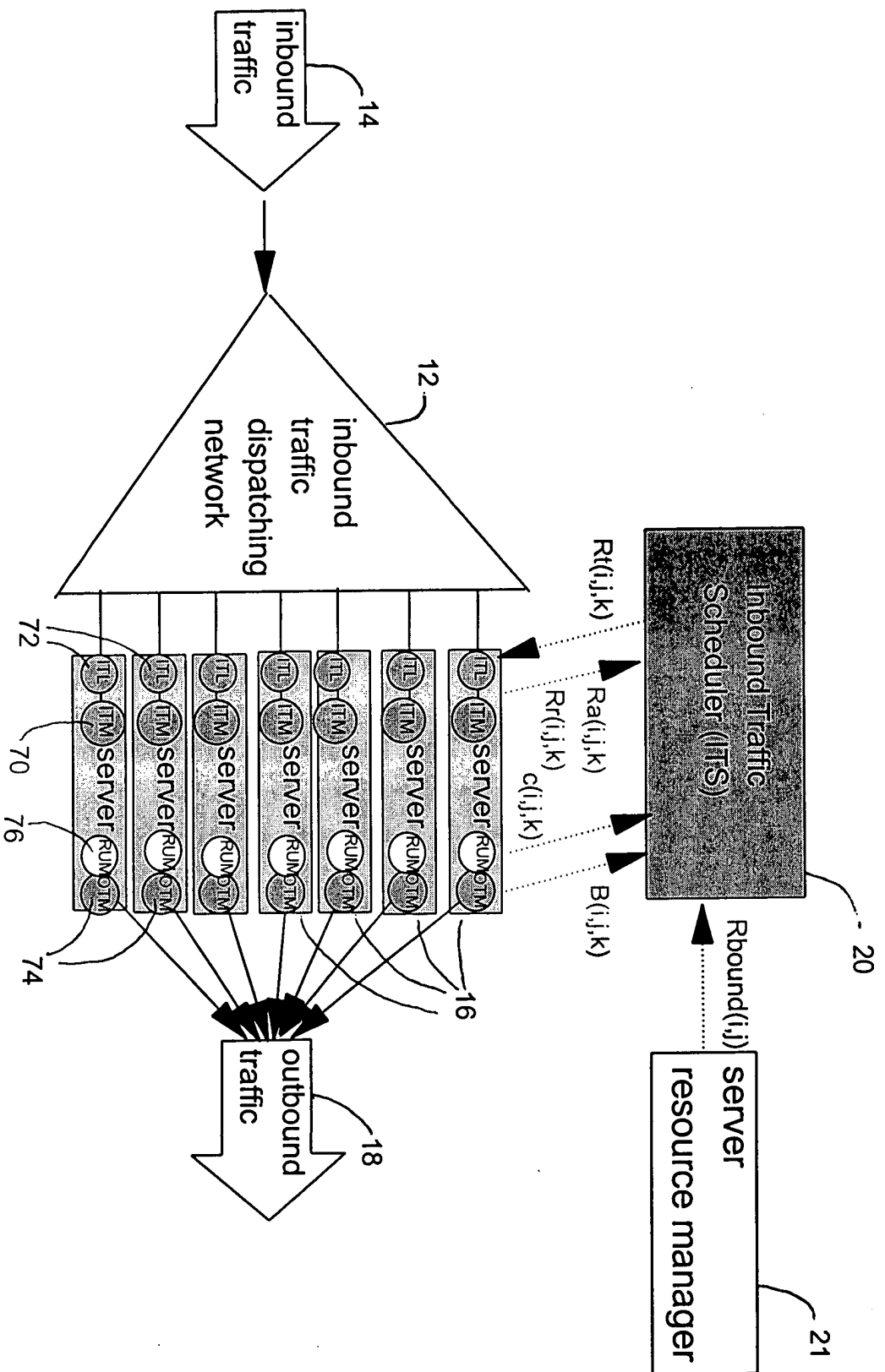


Fig. 7